

AMENDMENTS TO THE SPECIFICATION

Please amend the specification at the second paragraph on Page 1, starting at line 10 of page 1 and ending on line 5 of page 2, as follows:

A 1

The present invention is also related to two patent applications related to the selective locking of a keyboard. These patent applications, which are sometimes referred to as the ROM Scan Applications, are Serial No. 09/052,733 (issued on August 1, 2000 as U.S. Patent No. 6,098,171 B1), entitled "Personal Computer ROM Scan During Startup Protection" filed March 31, 1998 by Robert Duane Johnson et al. and "Method and System for Improved Security During ROM Scan," Serial No. 09/431,728 (issued on November 26, 2002 as U.S. Patent No. 6,487,465 B1) filed on November 1, 1999, by Richard Alan Dayan et al. The ROM Scan Applications are assigned to the assignee of the present invention, with the disclosures of these patents specifically incorporated herein by reference.

Please amend the full paragraphs of the specification, starting with Page 9, line 15, and ending on Page 13, line 20, as follows:

A 2

Figure 2 is a schematic diagram of a portion of the personal computer 10. The keyboard 14 is coupled through a keyboard/mouse controller 22 via a Low Pin Count (LPC) or ISA bus 24 (shown and labeled in Figure 3) to the I/O Controller Hub (ICH) 28, and via a Hub Link Bus (HLB) to a Memory Controller Hub (MCH) 27, and via a Front Side Bus (FSB) to the central processor 26 or the personal computer 10. Also connected to MCH 27 are a system memory 34 and an Accelerated Graphics Port (AGP) 32. ICH 28 includes a USB host controller 30, about which more is described in reference to Figure 3.

Access by the central processor [[10]] 26 is via the [[processors]] processor's address space at I/O address space addresses 60<sub>hex</sub> and 64<sub>hex</sub> 60<sub>hex</sub> and 64<sub>hex</sub>. The mouse 16 is also coupled to the keyboard/mouse controller 22. Both the keyboard 14 and mouse 16 ports are referred to as the PS/2 Keyboard and PS/2 Mouse

ports, respectively in the PC industry. As known in the state of the art, any device that emulates either a keyboard or mouse can attach to the respective port. In many personal computers, the keyboard 14 and mouse 16 ports are dedicated to their respective devices and are only configured to allow the attachment of such a device.

**Figure 3** is a schematic diagram illustrating the locking system of the present invention. The Keyboard/mouse controller 22, which is resident in the Super I/O module 29 and used to connect the keyboard 14 to the microprocessor 26 (shown in Figure 2), is connected to a security unit 82 which is a new connection for this invention. Alternative connections are possible to someone familiar with the state of the art.

For example, the security unit 82 could be connected to the LPC bus 24 [[and]] to monitor the transmissions for commands targeting the keyboard 14 or its controller 22.

*A2*

The USB host controller 30 is connected to the USB ports 88 via an interposing switch 80. The switch 80 receives instructions from the security unit 82 to instruct the switch 80 to lock or unlock the bus via a control signal [[89]] from security unit 82. When locked, the switch prevents data from reaching the USB host controller 30 and the microprocessor [[24]] 26 (shown in Figure 2). [[, however]] However, the USB Keyboard sensing unit 84 can still monitor the transmissions from devices attached to the USB ports 88 to monitor for entry of the password in order to unlock the bus 88. As USB keyboard keystrokes are detected, the keyboard sensing unit unpacks the USB usage codes and converts them to the well known PS/2 keyboard scan codes [[via bus 90]] to the security unit [[90]] 82 for correct password entry verification. When unlocked, the switch 80 allows all USB transmissions from devices attached to the USB ports 88 to the USB host controller 30 and the microprocessor [[24]] 26. In this way, when the switch 80 is in the locked state and keyboard inputs are not being processed from the USB ports 88 by the microprocessor 26, there is still something in the personal computer (the security unit 82) listening for a correct password to unlock the system and allow direct communication from either the keyboard 14 and/or a USB keyboard attached to one of the USB ports 88.

Figure 4 shows the logic in use in the USB Keyboard sensing unit 84 of this invention. The sensing unit 84 constantly monitors [[60]] (step 60) the USB [[bus]] ports 88 for the presence of data and commands.

If data is found, it is checked to see if it is a Control Request [[62]] (step 62). If not a control request, the data is checked to see if a USB device is sending data to the controller [[70]] (step 70). If it is not a data packet, the sensing unit 84 returns to monitor the USB bus [[60]] (step 60). If a USB data packet is present [[70]] (step 70), the sensing unit 84 checks to see if it is a keyboard device identified [[72]] (step 72) in step 68. If not a keyboard data packet, the sensing unit 84 returns to monitor the USB bus [[60]] (step 60). If it is a keyboard data packet [[72]] (step 72), the sensing unit detects the usage code from the data packet [[74]] (step 74) and converts the usage code to the industry standard scan code used by the PS/2 Keyboard device [[76]] (step 76). The sensing unit 84 then transmits the scan code (step 78) to the Security Unit 82 for processing and returns to step 60 to monitor the USB [[bus]] ports 88 for more data packets.

*A7*  
Returning to step 62, if the data is a control request, the sensing unit tests to see if it is a USB Keyboard Descriptor [[64]] (step 64). If not, the sensing unit returns to its monitoring state in step 60. If the data is a keyboard descriptor [[64]] (step 64), the sensing unit looks for an ID command [[66]] (step 66). When found, the USB ID is stored so that the USB device is recognized as a USB keyboard (step 68). Then processing returns to step 60 where the monitoring process resumes.

Figure 5 illustrates a logic design for the security unit 82 to allow it to recognize a correct password to unlock the keyboard attached to the system when the personal computer (and its processor 26) is otherwise locked against user inputs. The security unit 82 receives, at block 100, a single unit of data, such as would emanate from a single key stroke on a PS/2 personal computer keyboard or a USB keyboard attached at USB interface 88, indicating either a single character or a command from the processor 26 to the keyboard and checked to see if this data is a Load Password Command from the processor. If it is a Load Password Command, the security unit 82 intercepts and stores

the next set of characters as the password until a terminator (00h) is encountered [[102]] (block 102). Processing continues at step 100 again.

Returning to step 100, if the data is not a Load Password Command, the security unit 82 checks to see if the data is an Enable Password Command [[104]] (step 104) from the processor 26. If not, the security returns to step 100 to monitor the USB [[bus]] port 88 and PS/2 I/O ports 86 (60h and 64h) [[86]]. If the data is an Enable Password command, the security unit 82 checks to see if a valid password is already loaded [[106]] (step 106). If not, the security unit returns to step 100 to continue monitoring. If a valid password is already loaded, the security unit 82 locks the switch 80 in step 108. Following locking the keyboard, the security unit 82 goes into a monitoring state to check for the entry of a valid password [[110]] (step 110). The password may be entered on either the PS/2 Keyboard 14 or a USB keyboard attached at the USB interface 88. The system remains locked with respect to keyboard entry until the password is correctly entered. In step 112, the security unit 82 checks to see if the password was entered. If not entered correctly, the security unit 82 [[go]] goes to step 110 to monitor for entry of a password once again. If entered correctly, the switch 80 is unlocked [[114]] (step 114) and the security unit 82 [[start]] starts the process over again at step 100.

A2